

### **Remarks**

Reconsideration of the application is respectfully requested in view of the foregoing amendments and following remarks. Claims 1-23, 26-32, 34-37, 39-41, and 43 are pending in the application. No claims have been allowed. Claims 1, 15, 31, 34, 39, and 43 are independent. Claims 1, 2, 15, 26, 28, 31, 32, 34, 39, and 43 have been amended. Claims 24, 25, 33, 38, and 42 have been canceled without disclaimer and without prejudice to pursuing in a continuing application.

### **Initialed Form 1449 Not Complete**

On May 3, 2006, Applicants submitted an Information Disclosure Statement listing two non-patent references, in addition to a number of patent/application references. Applicants received the 1449 form, but the two non-patent references were not initialed. Applicants note that the non-patent references were submitted along with the IDS and are present in the Image File Wrapper. Applicants respectfully request that the Examiner provide an initialed 1449 form for this IDS submission. See MPEP § 609 (“An information disclosure statement filed in accordance with the provisions of 37 CFR 1.97 and 37 CFR 1.98 will be considered by the examiner assigned to the application.”).

### **Cited Art**

The Office action (“Action”) applies the following cited art: Michael G. Burke et al., “The Jalpeño dynamic optimizing compiler for Java™,” *Proceedings of ACM 1999 Conference on Java Grande*, ISBN: 1-58113-161-5, pp. 129-141, June 1999 (Burke); and Bates et al., U.S. Patent Application No. 2003/0145312 (Bates).

### **Interview Summary**

On January 17, 2008, Applicants representatives conducted a telephonic interview with the Examiner. Applicants argued that at no point did Burke teach or suggest a plurality of different languages (e.g., source or programming languages). This argument was viewed in a favorable light by the Examiner, and the language “plurality of different source languages” or “plurality of different programming languages” has been added to the independent claims.

### **Objections to the Specification**

The Action notes that the trademark "Microsoft" and the acronym "CLR" appear in the specification. Applicants recognize that "Microsoft" is a trademarked term. In order to clarify that "Microsoft" is a trademarked term, and to clarify the phrase "Microsoft CLR," Applicants have amended the specification to recognize the trademark, spell out the acronym, and include the "generic terminology" for the phrase "Microsoft CLR" as suggested by MPEP § 608.01(v). Specifically, Applicants have amended the specification to include the language "(the Common Language Runtime (CLR) is Microsoft's commercial implementation of the Common language Infrastructure (CLI) specification; Microsoft is a trademark of Microsoft Corporation)" after the first occurrence of the phrase "Microsoft CLR."

### **Objections to the Claims**

Claim 2 has been amended to spell out the acronym "JIT" as such: "lightweight just-in-time (JIT) compilers." Claim 28 has been amended to spell out the acronym "CDK" as such: "a compiler development kit (CDK)." Applicants respectfully request withdrawal of the Examiner's objections to the claims on this basis.

### **§ 103 Rejections**

The Action rejected claims 1-43 under 35 U.S.C. § 103(a) as unpatentable over Burke in view of Bates. Applicant respectfully submits the claims are allowable over the cited art. The rejections are traversed.

#### **Claim 1 is Allowable Over Burke in View of Bates**

Claim 1 recites one or more computer-readable media with computer-executable instructions for implementing a software development architecture comprising, in part:

one or more exception handling models operable to support a plurality of programming language specific exception handling models for a plurality of different source languages.

Burke does not teach or suggest the above recited language of claim 1.

Specifically, Burke does not teach or suggest a plurality of programming language specific exception handling models **for a plurality of different source languages**. Burke

describes the Jalapeno Optimizing Compiler which is a "dynamic optimizing compiler for **Java**." (Burke, Abstract, emphasis added.) The only language that the Jalapeno Optimizing Compiler is constructed for is compiling Java. Java bytecode is input to the Jalapeno Optimizing Compiler and then converted to an intermediate representation (IR). (Burke, FIG. 3.) To the extent that Burke describes exception handling models, Burke states that "[a] key difference between the Jalapeno [high-level intermediate representation] and Java bytecodes is the addition of separate operators to implement explicit checks for several common run-time exceptions . . ." (Burke, page 131, para. 2 of section 4.) However, as described above Burke describes a system that is operable only to compile **Java** bytecodes, and therefore only handles Java run-time exceptions. Thus, at no point does Burke teach or even suggest "one or more exception handling models operable to support a plurality of programming language specific exception handling models for a plurality of different source languages" as recited in claim 1.

Bates does not cure the above cited deficiencies of Burke because at no point does Bates teach or suggest "one or more exception handling models operable to support a plurality of programming language specific exception handling models for a plurality of different source languages" as recited in claim 1, nor does the Examiner allege that it does.

Thus, Burke and Bates, whether considered separately or in combination with each other, do not teach or suggest each and every element of claim 1. Claim 1 is therefore allowable, and Applicants respectfully request withdrawal of the § 103(a) rejection and allowance of claim 1.

#### Independent Claim 15 is Allowable over Burke in view of Bates

Claim 15 recites a method of creating a target software development tool, the method comprising, in part:

wherein the intermediate representation format comprises one or more exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages.

As discussed above with respect to claim 1, Burke does not teach or suggest "one or more exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages" as recited in independent claim 15.

To the extent that Burke describes exception handling models, Burke states that "[a] key difference between the Jalapeno [high-level intermediate representation] and Java bytecodes is the addition of separate operators to implement explicit checks for several common run-time exceptions . . ." (Burke, page 131, para. 2 of section 4.) Thus, Burke only describes a system that is operable to handle **Java** run-time exceptions.

Bates does not cure the above cited deficiencies of Burke because at no point does Bates teach or suggest "one or more exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages" as recited in independent claim 15, nor does the Examiner allege that it does.

Thus, Burke and Bates, whether considered separately or in combination with each other, do not teach or suggest each and every element of independent claim 15. Independent claim 15 is therefore allowable, and Applicants respectfully request withdrawal of the § 103(a) rejection and allowance of independent claim 15.

Independent Claim 31 is Allowable over Burke in view of Bates

Independent claim 31 recites a method of creating a target software development tool from a common framework, the method comprising, in part:

wherein the common framework comprises exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages.

As discussed above with respect to claim 1, Burke does not teach or suggest "exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages" as recited in independent claim 31.

To the extent that Burke describes exception handling models, Burke states that "[a] key difference between the Jalapeno [high-level intermediate representation] and Java bytecodes is the addition of separate operators to implement explicit checks for several common run-time exceptions . . ." (Burke, page 131, para. 2 of section 4.) Thus, Burke only describes a system that is operable to handle **Java** run-time exceptions.

Bates does not cure the above cited deficiencies of Burke because at no point does Bates teach or suggest "exception handling models capable of supporting a plurality of programming language-specific exception handling models for the plurality of different programming languages" as recited in independent claim 31, nor does the Examiner allege that it does.

Thus, Burke and Bates, whether considered separately or in combination with each other, do not teach or suggest each and every element of independent claim 31. Independent claim 31 is therefore allowable, and Applicants respectfully request withdrawal of the § 103(a) rejection and allowance of independent claim 31.

Independent Claim 34 is Allowable over Burke in view of Bates

Independent claim 34 recites a method of producing inter-compatible software development tools, the method comprising in part:

wherein the software development architecture comprises functionality for exception handling models operable to support programming-language specific exception handling models for the plurality of different programming languages, and the software development architecture is used by both the first and second software development tools.

Burke does not teach or suggest the above cited language of independent claim 34.

As discussed above with respect to claim 1, Burke does not teach or suggest "exception handling models operable to support programming-language specific exception handling models for the plurality of different programming languages" as recited in independent claim 34.

To the extent that Burke describes exception handling models, Burke states that "[a] key difference between the Jalapeno [high-level intermediate representation] and Java bytecodes is the addition of separate operators to implement explicit checks for several common run-time exceptions . . ." (Burke, page 131, para. 2 of section 4.) Thus, Burke only describes a system that is operable to handle **Java** run-time exceptions.

Bates does not cure the above cited deficiencies of Burke because at no point does Bates teach or suggest "exception handling models operable to support programming-language specific exception handling models for the plurality of different programming languages" as recited in independent claim 34, nor does the Examiner allege that it does.

Thus, Burke and Bates, whether considered separately or in combination with each other, do not teach or suggest each and every element of independent claim 34. Independent claim 34

is therefore allowable, and Applicants respectfully request withdrawal of the § 103(a) rejection and allowance of independent claim 34.

Independent Claim 39 is Allowable over Burke in view of Bates

Independent claim 39 recites a method of modifying a software development tool, the software development tool having been created using a software development architecture that is operable for a plurality of different programming languages and comprising a common framework operable to support a plurality of languages and further comprising one or more software development components, the method comprising, in part:

wherein the binary version of the software development architecture comprises functionality for exception handling models operable to support a plurality of programming language specific exception handling models for the plurality of different programming languages used by the modified software development tool.

Burke does not teach or suggest the above cited language of independent claim 39.

As discussed above with respect to claim 1, Burke does not teach or suggest "exception handling models operable to support a plurality of programming language specific exception handling models for the plurality of different programming languages" as recited in independent claim 39.

To the extent that Burke describes exception handling models, Burke states that "[a] key difference between the Jalapeno [high-level intermediate representation] and Java bytecodes is the addition of separate operators to implement explicit checks for several common run-time exceptions . . ." (Burke, page 131, para. 2 of section 4.) Thus, Burke only describes a system that is operable to handle **Java** run-time exceptions.

Bates does not cure the above cited deficiencies of Burke because at no point does Bates teach or suggest "exception handling models operable to support a plurality of programming language specific exception handling models for the plurality of different programming languages" as recited in independent claim 39, nor does the Examiner allege that it does.

Thus, Burke and Bates, whether considered separately or in combination with each other, do not teach or suggest each and every element of independent claim 39. Independent claim 39 is therefore allowable, and Applicants respectfully request withdrawal of the § 103(a) rejection and allowance of independent claim 39.

Independent Claim 43 is Allowable over Burke in view of Bates

Independent claim 43 recites a method of creating a software development tool, the method comprising, in part:

one or more exception handling models capable of supporting a plurality of programming language specific exception handling models for the plurality of different programming languages.

Burke does not teach or suggest the above recited language of claim 43.

As discussed above with respect to claim 1, Burke does not teach or suggest "one or more exception handling models capable of supporting a plurality of programming language specific exception handling models for the plurality of different programming languages" as recited in independent claim 43.

To the extent that Burke describes exception handling models, Burke states that "[a] key difference between the Jalapeno [high-level intermediate representation] and Java bytecodes is the addition of separate operators to implement explicit checks for several common run-time exceptions . . ." (Burke, page 131, para. 2 of section 4.) Thus, Burke only describes a system that is operable to handle **Java** run-time exceptions.

Bates does not cure the above cited deficiencies of Burke because at no point does Bates teach or suggest "one or more exception handling models capable of supporting a plurality of programming language specific exception handling models for the plurality of different programming languages" as recited in independent claim 43, nor does the Examiner allege that it does.

Thus, Burke and Bates, whether considered separately or in combination with each other, do not teach or suggest each and every element of independent claim 43. Independent claim 43 is therefore allowable, and Applicants respectfully request withdrawal of the § 103(a) rejection and allowance of independent claim 43.

Dependent Claims 2-14, 16-23, 26-30, 32, 35-37, 40, and 41 are Allowable over Burke in view of Bates

Dependent claims 2-14, 16-23, 26-30, 32, 35-37, 40, and 41, which each depend directly or indirectly from one of the independent claims discussed above, are allowable over Bates in

view of Burke for at least the reasons above. Applicants note that the dependent claims may also be allowable for reasons other than those discussed above with regards to their respective independent claims. Applicants respectfully request withdrawal of the § 103(a) rejection and allowance of dependent claims 2-14, 16-23, 26-30, 32, 35-37, 40, and 41.

#### **Request for Interview**

If any issues remain, the Examiner is formally requested to contact the undersigned attorney prior to issuance of the next Office action in order to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the foregoing formal Amendment so that the Examiner may fully evaluate Applicants' position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

#### **Conclusion**

The claims should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Telephone: (503) 595-5300  
Facsimile: (503) 595-5301

By /Cory A. Jones/  
Cory A. Jones  
Registration No. 55,307